5   We claim:

1.   A method for enhancing flow analysis, comprising:

inferring types from a program;

forming a type graph having polarities and indices from the types; and

10   forming a flow graph from the type graph to inhibit imprecise paths so as

to enhance context-sensitivity of flow analysis.

2.   The method of claim 1, wherein inferring includes generating constraints

from the program and solving the constraints.

15

3.   The method of claim 2, wherein generating includes generating the

constraints, wherein the constraints include a set of equalities and inequalities,

wherein the set is a finite set, and wherein the set of equalities and inequalities is

adapted to be a set of simultaneous equations.

20

4.   The method of claim 3, wherein generating includes generating the

constraints, wherein an equality from the set of equalities defines that a type is

equal to another type, and wherein the type and the another type are adapted to be

unified.

25

5   5.   The method of claim 3, wherein generating includes generating the

constraints, wherein an inequality from the set of inequalities defines that a type is

an instance of another type, wherein the inequality includes an index and a

polarity.

10   6.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for enhancing flow analysis, the method

comprising:

        inferring types from a program;

        forming a type graph having polarities and indices from the types; and

15   forming a flow graph from the type graph so as to enhance context-

sensitivity of flow analysis.

7.   A method for enhancing flow analysis, comprising:

        forming a type graph that includes polarities and indices; and

20   forming a flow graph that includes a set of flow paths, wherein the set of

flow paths excludes imprecise paths so as to enhance context-sensitivity of flow

analysis.

8.   The method of claim 7, wherein forming the set of flow paths includes

25   forming at least one flow path that inherits a polarity from the polarities of the

5   type graph.

9.   The method of claim 8, wherein forming includes forming the set of flow

paths, wherein each imprecise path includes two flow edges, wherein the polarity

of one of the two flow edges is negative, and wherein the polarity of the other of

10   the two flow edges is positive.

10.   The method of claim 9, wherein forming includes forming the set of flow

paths, wherein each imprecise path is defined by the one of the two flow edges

preceding the other of the two flow edges.

15

11.   The method of claim 7, wherein forming includes forming the type graph,

wherein each polarity indicates at least one of a value flowing into a type and a

value flowing out of a type.

20   12.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for enhancing flow analysis, the method

comprising:

forming a type graph that includes polarities and indices; and

forming a flow graph that includes a set of flow paths, wherein the set of

25   flow paths excludes imprecise paths so as to enhance context-sensitivity of flow

5 analysis.

13.   A method for forming a graph to enhance flow analysis, comprising:

forming a plurality of nodes to represent types; and

forming a plurality of instantiation paths to represent instantiations of

10   generic types to instance types, wherein each instantiation path includes an index

and a polarity.

14.   The method of claim 13, further comprising forming a plurality of flow

paths to represent a flow of values between types.

15

15.   The method of claim 14, wherein forming includes forming a plurality of

flow paths, wherein each flow path includes a polarity that is inherited from an

instantiation path.

20   16.   The method of claim 15, wherein forming includes forming a plurality of

flow paths, wherein the polarity defines at least one of an ingress and an egress

flow of a value with respect to a type.

17.   The method of claim 13, wherein forming includes forming a plurality of

25   instantiation paths, wherein the index defines an occurrence of an instantiation of a

5   generic type to an instance type so as to differentiate among occurrences of

instantiations of the generic type to the instance type.

18.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for forming a graph to enhance flow

10   analysis, the method comprising:

forming a plurality of nodes to represent types; and

forming a plurality of instantiation paths to represent instantiations of

generic types to instance types, wherein each instantiation path includes an index

and a polarity.

15

19.   A method for enhancing flow analysis, comprising:

abstracting program expressions into types; and

forming a type graph from the types, wherein the type graph includes

polarities and indices so as to enhance flow analysis.

20

20.   The method of claim 19, wherein forming includes forming a plurality of

nodes, wherein each node represents a type.

21.   The method of claim 20, wherein forming includes forming at least one

25   instantiation path between two nodes to represent an instantiation of a generic type

5   to an instance type.

22.   The method of claim 21, wherein forming includes forming at least one

instantiation path that includes an index and a polarity.

10   23.   The method of claim 19, wherein abstracting includes unifying recursive

types, wherein unifying is adapted to be executed using a cyclic unification

technique.

24.   A computer readable medium having instructions stored thereon for

15   causing a computer to perform a method for enhancing flow analysis, the method

comprising:

abstracting program expressions into types; and

forming a type graph from the types, wherein the type graph includes

polarities and indices so as to enhance flow analysis.

20

25.   A method for inferring types to enhance flow analysis, comprising:

generating constraints from a program; and

solving the constraints to infer at least one type, wherein solving includes

propagating polarities so as to enhance flow analysis.

25

5   26.   The method of claim 25, wherein propagating includes propagating

polarities, wherein each polarity is adapted to indicate at least one of a negative

polarity, a positive polarity, and a bidirectional polarity.

27.   The method of claim 25, wherein propagating includes propagating

10   polarities, wherein the negative polarity defines a value flowing into a type.

28.   The method of claim 25, wherein propagating includes propagating

polarities, wherein the positive polarity defines a value flowing out of a type.

15   29.   The method of claim 25, wherein propagating includes propagating

polarities, wherein the bi-directional polarity defines a combination of a negative

polarity and a positive polarity.

30.   A computer readable medium having instructions stored thereon for

20   causing a computer to perform a method for inferring types to enhance flow

analysis, the method comprising:

generating constraints from a program; and

solving the constraints to infer at least one type, wherein solving includes

propagating polarities so as to enhance flow analysis.

25

5   31.   A method for enhancing flow analysis, comprising:

annotating each expression in a program by a label;

associating the label of an expression with a type of the expression; and

tracing at least one path on a type graph having polarities to determine if a

value arising at one label in the program flows to another label in the program.

10

32.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for enhancing flow analysis, the method

comprising:

annotating each expression in a program by a label;

15   associating the label of an expression with a type of the expression; and

tracing at least one path on a type graph having polarities to determine if a

value arising at one label in the program flows to another label in the program.

33.   A method for enhancing flow analysis, comprising:

20   forming a type instantiation graph that includes polarities and indices; and

computing points-to information for at least one program point by

answering reachability queries on the type instantiation graph.

34.   The method of claim 33, wherein forming includes forming with a

25   polymorphic inference technique based on instantiation constraints.

5

35.   The method of claim 33, wherein answering includes answering

reachability queries within a desired duration, wherein the desired duration is

linearly proportional to a size of the type instantiation graph.

10   36.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for enhancing flow analysis, the method

comprising:

        forming a type instantiation graph that includes polarities and indices; and

        computing points-to information for at least one program point by

15   answering reachability queries on the type instantiation graph.

37.   A data structure to enhance flow analysis, comprising:

        a data member type to represent a type of a program expression; and

        a data member flow having a data member polarity and a data member

20   index to represent a flow path between two types.

38.   The data structure of claim 37, wherein the data member type is adapted to

contain at least one of a generic type and an instance type.

25   39.   The data structure of claim 37, wherein the data member type is adapted to

5   contain an instance type that is an instantiation of a function type.

40.   The data structure of claim 37, wherein the data member type is adapted to contain an instance type that is an instantiation of a pointer type.

10   41.   The data structure of claim 37, wherein the data member flow contains an address of another type so that the data member flow represents a flow path between the data member type and the another type.

42.   A method for enhancing flow analysis, comprising:

15   forming a type instantiation graph that includes polarities and indices;

           forming a flow graph to form a set of flow paths; and

           forming a subset from the set of flow paths such that context-sensitivity of flow analysis is enhanced.

20   43.   The method of claim 42, wherein forming includes forming a set of flow paths that includes at least one imprecise path, wherein the subset excludes the at least one imprecise path.

44.   The method of claim 43, wherein forming the type-instantiation graph

25   includes forming a plurality of nodes, wherein each node of the plurality of nodes

5   represents a type expression.

45.   The method of claim 44, wherein forming includes forming a plurality of

instantiation paths, wherein each instantiation path represents an instantiation of a

generic type expression to an instance type expression, wherein each instantiation

10   path includes an index and a polarity, wherein the index represents an occurrence

of an instantiation of the generic type expression to the instance type expression.

46.   The method of claim 45, wherein forming includes forming a set of flow

paths such that each flow path includes a polarity, wherein the polarity of the flow

15   path is positive if the polarity of the instantiation path is positive, and wherein the

polarity of the flow path is negative if the polarity of the instantiation path is

negative.

47.   The method of claim 46, wherein forming a set of flow paths includes

20   forming a path, wherein the path includes at least two flow edges.

48.   The method of claim 47, wherein forming a path includes forming a path,

wherein the path is defined to be an imprecise path if a polarity of one of the at

least two flow edges is negative and a polarity of an other of the at least two flow

25   edges is positive, and wherein the one of the at least two flow edges precedes the

5   other of the at least two flow edges.

49.   A computer readable medium having instructions stored thereon for

causing a computer to perform a method for enhancing flow analysis, the method

comprising:

10   forming a type instantiation graph that includes polarities and indices;

           forming a flow graph to form a set of flow paths; and

           forming a subset from the set of flow paths such that context-sensitivity of

flow analysis is enhanced.

15   50.   A graph for enhancing program analysis, the graph comprising:

           a plurality of nodes that represent type expressions;

           a plurality of instantiation lines that represent type instantiations; and

           a plurality of flow lines, wherein each flow line represents a flow direction

of at least one value.

20

51.   The graph of claim 50, wherein each instantiation line emanates from one

node that is indicative of a generic type expression and ends at another node that is

indicative of an instance type expression to define an instantiation direction.

25   52.   The graph of claim 50, wherein each flow line includes a polarity, wherein

5   each flow line emanates from one node that is indicative of the source of the at

least one value and ends at another node that is indicative of the target of the at

least one value to define a flow direction.

53.   The graph of claim 52, wherein each instantiation line includes an

10   instantiation constraint.

54.   The graph of claim 53, wherein the instantiation constraint includes an

index and a polarity, wherein the index represents an occurrence of an

instantiation of the generic type to the instance type.

15

55.   The graph of claim 54, wherein the polarity of the flow line is positive if

the polarity of the instantiation line is positive, wherein the polarity of the flow

line is negative if the polarity of the instantiation line is negative.

20   56.   The graph of claim 54, wherein the flow direction of the flow line is the

same as the instantiation direction of the instantiation line if the polarity of the

instantiation line is positive, wherein the flow direction of the flow line is opposite

the instantiation direction of the instantiation line if the polarity of the instantiation

line is negative.